Mobile-to-Mobile Video Recommendation

Padmanabha Venkatagiri Seshadri, Mun Choon Chan, and Wei Tsang Ooi

National University of Singapore, Computing 1, 13 Computing Drive 117417, Singapore
{padmanab,chanmc,ooiwt}@comp.nus.edu.sg

Abstract. Mobile device users can now easily capture and socially share video clips in a timely manner by uploading them wirelessly to a server. When attending crowded events, however, timely sharing of videos becomes difficult due to choking bandwidth in the network infrastructure, preventing like-minded attendees from easily sharing videos with each other through a server. One solution to alleviate this problem is to use direct device-to-device communication to share videos among nearby attendees. Contact capacity between two devices, however, is limited, and thus a recommendation algorithm is needed to select and transmit only videos of potential interest to an attendee. In this paper, we address the question: which video clip should be transmitted to which user. We proposed an video transmission scheduling algorithm, called *CoFiGel*, that runs in a distributed manner and aims to improve both the prediction coverage and precision of the recommendation algorithm. At each device, CoFiGel transmits the video that would increase the estimated number of positive user-video ratings the most if this video is transferred to the destination device. We evaluated *CoFiGel* using real-world traces and show that substantial improvement can be achieved compared to baseline schemes that do not consider rating or contact history.

Keywords: mobile-to-mobile communication, memory based collaborative filtering, coverage.

1 Introduction

Mobile devices are increasingly capable in their abilities to sense, capture, and store rich multimedia data. Multiple wireless interfaces facilitate users to upload and share their experience with friends and public. In this work, we are interested in mobile video sharing among attendees of an event. As an example, consider product exhibitions, malls, museums, game events such as the Olympics, where people have to move around in a large area and could benefit from receiving video clips of a small portion of the event so that they can decide whether to attend it.

This information sharing paradigm emphasizes both spatial locality and timeliness and is different from archived video sharing services such as those provided by YouTube. A straight forward approach to enable such video sharing is to have users upload the videos captured to a central server through 3G/HSPA networks. Users can then search for or browse through the uploaded videos through the server. While this approach can provide good performance in terms of delivering the right videos to the right users, it has obvious drawbacks. First, when user density is high, there is likely to be insufficient aggregate upload bandwidth for the combination of large amount of data

K. Zheng, M. Li, and H. Jiang (Eds.): MOBIQUITOUS 2012, LNICST 120, pp. 13–24, 2013.

and large number of users. Next, the use of 3G/HSPA network for upload is relatively inefficient, since the network is optimized for download. Finally, videos stored in the central server have to be downloaded to individual mobile phone for viewing and rating, further straining the 3G/HSPA network.

The approach adopted in this work is to circumvent the cellular network infrastructure and transfer videos directly, from one mobile device to another mobile device, via short range connection such as WiFi or Bluetooth. A user captures a video and indicates the mobile application to share it. The video is pushed to nearby devices when connections to these devices becomes available. A user can choose to watch and rate videos accumulated in the video inbox of the device. The device can also forward the videos through a mobile-to-mobile network.

Besides alleviating the network infrastructure bottleneck, direct mobile-to-mobile communication may also reduce power consumption [9]. In addition, the much shorter RTT for direct mobile-to-mobile transfer allows significantly higher throughput compared to transferring large amount of data over the Internet through the 3G/HSPA network, where the median ping latency has been observed to be almost 200ms [2].

The use of short range communication among mobile devices results in intermittent connectivity. These devices, in essence, form a delay-tolerant network (DTN). As mobile devices have limited contact time, pushing the right video to a neighboring device is especially important. Ideally, we want videos that a user is interested in to end up in its inbox within a given time period. As such, our system uses a collaborative filtering (CF) based recommender system to predict the user preference. The use of this algorithm, however, requires collection of sufficient number of user-item ratings to work. In other words, pushing a video to a user now has two purposes: for the user to watch and for the user to rate. The decision to select which video to transfer should thus consider the needs of the CF algorithm as well.

To address this challenge, we propose *CoFiGel*, a video transfer scheduler in the DTN context that integrates CF-based recommender system. CoFiGel effectively utilizes the limited contact capacity among mobile devices to filter and disseminate user-generated videos published by mobile users to other mobile users. CoFiGel is designed to (i) increase the prediction coverage, which is the ability of the algorithm to predict ratings for items, and (ii) route videos in such a way that increases the item precision, i.e., the percentage of items recommended to users that are rated positively.

We evaluate CoFiGel through trace-based simulation using RollerNet human mobility trace [6] and an user rating data set from MovieLens[1]. Our evaluation shows that CoFiGel can provide 80% more prediction coverage in comparison to the baseline algorithms, detecting at least 74% of positive ratings in the process, and delivers at least 59% more positive (liked by user) items in comparison to the baseline algorithms that do not take into account either ratings or contact history.

The rest of the paper is organized as follow. Section 2 discusses related work. Section 3 describes our mobile-to-mobile video transfer application and motivates the need for CoFiGel. CoFiGel is presented in Section 4 and is evaluated in Section 5. Finally, Section 6 concludes.

[1] MovieLens Dataset, `http://www.grouplens.org/node/73`

2 Related Work

Collaborative Filtering (CF). The most prominent and popular recommendation technique that has seen extensive research and wide deployment is collaborative filtering (CF) [3]. CF techniques can be broadly categorized into memory-based or model-based. Memory-based CF (MCF) utilizes rating history of users to identify neighbourhood patterns among users or items. This pattern facilitates the prediction of ratings for hitherto unrated user-item pairs. Model-based CF uses the user ratings in conjunction with standard statistical models such as Bayesian belief nets and latent semantic model to identify patterns in the ratings of user-item pairs. The resultant model is then used to make predictions for future ratings. There exists much research on using CF on peer-to-peer (P2P) systems. PocketLens [14] is a recommender system for portable devices that uses item-item collaborative filtering for making recommendations. It proposes a rating exchange protocol for both distributed P2P architecture and centralized server architecture, where nodes rely on a central server for storing rating information. A probabilistic model-based CF is proposed by Wang et al. [8] for a P2P network. Other related work focuses on the security and privacy aspect, including providing user incentive [5], trust of rating protocol and privacy [7].

DTN Content Dissemination. There are many unicast DTN routing schemes designed to improve point-to-point delivery probability and/or minimize delay [15]. These protocols, however, do not address the issue of information dissemination. A common problem studied in DTN content dissemination is how to maximize the freshness of dynamic content [10]. A subset of mobile devices download internet content and exchange among themselves so as to maximize freshness. Caching schemes where nodes refresh/reshuffle their cached content based on a voting process can also be exploited, as done by Ioannidis et al. [10]. In [11], predefined preferences are used to route items to users. However, preferences are static and not predicted. Another approach for content discovery and dissemination in DTN uses tags [12]. Tag metadata is propagated in the network and user interest is determined by matching tags.

Unlike previous work, *CoFiGel* provides a framework to integrate MCF and DTN routing, focusing on utilizing limited contact capacities in DTN to improve rating coverage and item recall. *CoFiGel* does not assume any specific MCF algorithm. Instead, it defines an abstract model of how MCF works and how the MCF should interact with the DTN routing protocol. We are not aware of any MCF that specifically takes into account the intermittent contact capacities of mobile nodes, nor any DTN mechanism that takes into account usefulness of item transferred to improve coverage and item recall of the MCF algorithm.

3 Mobile-to-Mobile Video Sharing

Motivation. We now motivate our work by demonstrating the efficacy and advantages of mobile-to-mobile video transfer.

Mobile data usage has outgrown available bandwidth in 3G/HSPA network, resulting in severe congestion in the cellular network in some cases[2]. A popular approach to reduce such congestion is to offload data traffic to the WiFi network whenever possible[3]. Communication over WiFi also consumes less power than 3G/HSPA network (four to six times less power for file transfer [9]). We further measure the performance of file transfer between a mobile device and a central server using 3G/HSPA network and between two mobile devices directly using WiFi.

To quantify the performance of mobile-to-server transfer, we upload and then download a 14.3 MB video clip to YouTube using a HSPA network, which provides a maximum download and upload rate of 7.2 Mbps and 1.9 Mbps respectively. The average download and upload throughputs measured (average of 5 trials) are 1125.2 kbps and 57 kbps respectively. To quantify the performance of mobile-to-mobile transfer, we use two Samsung Nexus S phones that support IEEE 802.11n (link rate is 72.2 Mbps) to exchange the same video file directly over a TCP connection. The measured throughput is 22.6Mbps (average of 5 trials). The 20-fold difference in measured throughput can be attributed to the differences in link rate and RTT observed (70ms for mobile-to-server and 5.5ms for mobile-to-mobile). This superior throughput motivates our study on mobile-to-mobile video sharing.

Mobile-to-Mobile Video Sharing. A user of mobile-to-mobile video sharing can share video content either generated or already available on the mobile device through a *video outbox* and watch and/or rate video available in a *video inbox*. When a device is within communication range through WiFi or Bluetooth, the scheduler uses the MCF to choose the subset of videos (interesting to the user) in the inbox to transfer within the limited contact time. The scheduler also manages the limited inbox space. Each device maintains a user-video rating matrix, which is updated either when a video is rated on the device, or when the device receives a rating matrix from another device. The rating matrix is one of the two meta-data (the other is contact history among devices) being exchanged between two devices when devices make contact with each other. Upon update of the matrix, predictions of interest-level of videos are recomputed and video transfers are rescheduled accordingly.

Memory-Based Collaborative Filtering. We now detail how MCF works. MCF is a class of recommender algorithms that is model independent and is able to capture the abstract user preference on a set of items. Typical MCF techniques have the following structure. A training data set is used to build a rating matrix consisting of ratings given for items by users. The rating matrix is used to identify the similarities between users-items and to predict the ratings of hitherto unrated items by a given user. Items that are predicted to have high ratings are shown to the user; Feedback from the user on these items is then used to update the rating matrix. The assumption is that *users tend to behave in the same way as they behaved in the past.*

[2] http://www.fiercewireless.com/europe/story/o2-germany-admits-network-meltdown-smartphones-blamed/2011-11-23

[3] http://www.telecomasia.net/content/3g-wifi-offload-pipes-singapore

Table 1. Rating matrix for Cosine-based similarity metric, $\diamond$ denotes ratings that could be predicted and $\star$ denotes unknown ratings

Users	i_1	i_2	i_3	i_4	i_5	i_6
u_1	1	$\diamond$	$\star$	$\star$	$\diamond$	$\diamond$
u_2	1	$\diamond$	$\star$	$\star$	$\diamond$	$\diamond$
u_3	1	1	$\star$	$\diamond$	$\diamond$	1
u_4	$\diamond$	1	$\diamond$	1	1	1
u_5	$\star$	$\diamond$	1	1	$\diamond$	$\diamond$
u_6	1	$\diamond$	$\star$	$\diamond$	1	1
u_7	1	0	$\star$	$\diamond$	0	1

For concreteness, we will use the Cosine-based similarity metric ([13]) in the rest of this paper to illustrate how CoFiGel works. Cosine-based similarity is a popular item-based MCF and has been used in large scale real-world applications such as the recommendation system used by *Amazon.com*. Note that CoFiGel can also work with other MCF algorithms, such as Slope One [3]. Since MCF works for recommendation of any kind of items, we will use the term *items* in the rest of the discussions to refer to videos in our application.

In general, ratings can be represented as integer values. For simplicity, we assume that ratings are binary and are expressed as either 1 (positive/like) or 0 (negative/dislike). In computing Cosine-based similarity, unrated items are assigned ratings of 0. After a user has rated an item, the item will not be recommended to the user again.

Let U and I be the set of all users and items respectively and I_u^+ and $I_u^?$ be the set of items that are rated positive and unrated by a user $u \in U$ respectively. Let the actual rating of an item $i \in I$ for user u be $r_{u,i}$. Cosine-based similarity metric computes $R_{u,i}$, the *rank* of an unrated user-item pair (u, i), in the following way. First, the similarity between two items i and j is computed using $Sim(i,j) = \left(\frac{\sum_{u \in U} r_{u,i} \cdot r_{u,j}}{\sqrt{\sum_{u \in U} r_{u,i}^2} \cdot \sqrt{\sum_{u \in U} r_{u,j}^2}} \right)$ and $R_{u,i} = \sum_{j \in I_u^+} Sim(i,j)$ Obviously, the rank of item i for user u can be computed only if there is at least one user who has rated both i and another item that user u has rated positively. If the rank cannot be computed, then we say that the particular user-item pair is *unpredictable*.

Table 1 shows a rating matrix with items that are rated (positively and negatively), predicted and unpredictable. Typically, the top-k items $i \in I_u^?$ with highest rank are recommended to user u. We say that the prediction of i is *positive* for u if i is among the top-k items in $I_u^?$, and *negative* otherwise. A prediction of i is said to be *correct*, if the predicted rating is consistent with the user rating eventually. Note that the notion of whether a prediction is positive or not changes over time (and thus whether it is correct or not changes over time as well).

The performance of MCF algorithm is measured by several standard metrics [3]. For instance, *precision* and *recall* are used to measure the classification performance of a MCF algorithm. Precision is a measure of recommended items that are relevant to the users, and recall is a measure of the number of relevant items that are recommended to the users. Another common performance measure used is *prediction coverage*, (or coverage for short), defined as the percentage of *the number of predictable user-item pair*.

MCF for Mobile-to-Mobile Recommendation. When two mobile devices meet, they need to select which items to be transmitted over the intermittent contacts based on the meta-data information available. As mentioned, since contact capacity is precious, items that are likely to be liked by other users should be transfer and propagated with higher priorities. Running MCF in the context of mobile-to-mobile video sharing, however, leads to another issue: since each user is likely get a chance to rate only a small subset of all videos available, selecting which items for users to rate is also important, to increase the coverage.

For the rating matrix shown in Table 1, item i_1 has three common user ratings with items i_2, i_5 and i_6. i_3 has only one common user rating with i_4. Using Equations for $Sim(i,j)$ and $R_{u,i}$, we can compute R_{u_4,i_1} and R_{u_4,i_3} as follows: $R_{u_4,i_1} = Sim(1,2)+Sim(1,4)+Sim(1,5)+Sim(1,6) = \frac{1}{\sqrt{5}\sqrt{2}} + 0 + \frac{1}{\sqrt{5}\sqrt{2}} + \frac{3}{\sqrt{5}\sqrt{4}} \approx 1.30$ and $R_{u_4,i_3} = Sim(3,2)+Sim(3,4)+Sim(3,5)+Sim(3,6) = 0 + \frac{1}{\sqrt{1}\sqrt{2}} + 0 + 0 \approx 0.71$. i_1 has a higher rating than i_3 with respect to u_4.

The coverage consideration, however, is different. It can be observed from Table 1 that all users except u_4 and u_5 have already rated i_1. Knowing the value of r_{u4,i_1}, allows only at most one more rating, R_{u_5,i_1}, to be computed. The gain in rated and predictable items is 2. On the other hand, i_3 has been rated only by u_5. Knowing the value of r_{u_4,i_3}, allows the rating of 3 users (u_3, u_6 and u_7) for item i_3 to be computed. The gain in rated and predictable items is 4. Therefore, the rating of i_3 by u_4 has a higher gain in rated and predictable items than rating i_1.

This example illustrates the trade-off between improving user satisfaction and improving coverage when not all data transfer can be completed within a contact. If user satisfaction is more important, then i_1 will be chosen for transfer. If coverage has higher priority, then i_3 should be chosen. Note that when there is a centralized server with continuous connectivity to users and has access to all rating information and data items, the impact of this trade-off is not significant. Such a trade-off, however, plays an important role in a resource constraint environment where the contacts between mobile devices are intermittent, contact capacities are limited and only subsets of data items can be stored in the local buffers. *The execution of MCF on mobile devices with intermittent contacts presents a new challenge that is not present in traditional application of MCF in a centralized or peer-to-peer environment where connectivities are not intermittent.*

4 CoFiGel

The MCF algorithm runs locally on each mobile device based on available meta-data information, which consists of the user-item rating matrix and contact history. We denote the element $m_{u,i}$ as the rating of item i by user u at any given time. The status of $m_{u,i}$ can be either **rated**, **predicted** or **unpredictable**. A rating $m_{u,i}$ is rated if i has be transferred to and rated by u, and the rating can be either 1 or 0. A rating $m_{u,i}$ is predicted if it has not been rated yet, but the rank $R_{u,i}$ (see section 3) can be computed. The predicted rating is 1 if i is among the top-k item according to $R_{u,i}$ for user u, and 0 otherwise. A rating $m_{u,i}$ is said to be *correct* if the predicted rating matches the user rating eventually.

Recall that there are two naive methods to pick an item to transfer to another device. The first method, considering only item recall, picks a predicted item that gives the highest rank $R_{u,i}$ to maximize the probability that the rating $m_{u,i}$ is correct and positive. The second method considers only the prediction coverage, and picks a predicted item such that *if* the item is rated, then the number of unpredictable items becoming predictable is maximal. To consider both recall and coverage, we consider the following metric: for an item i, we are interested in the number of correct positive prediction for i eventually, i.e., when i has been rated by all users. Before i is rated by all users, this quantity is considered as a random variable, denoted as Ω_i. At any round t, we know the current number of correct positive rating for i, denoted r_i^+. We also know is the number of positive predictions for item i, g_i^+. Ideally, we would like the following inequality to be true: $\Omega_i > r_i^+ + g_i^+$, i.e., all the positive predictions for i are correct, and there are additional new positive ratings for i. The key question is thus to estimate the probability that the above condition is true if i is transferred.

In the following, we present approximations on the potential positive ratings for an item and the probability of delivery of items with positive ratings to the users. The goal is to derive approximations that can be used as input to guide and motivate the design of *CoFiGel*.

Let, n be number of users, g_i^+ number of positive prediction for item i currently, r_i^+ number of correctly predicted positive ratings for item i currently, Ω_i random variable for number of correct positive ratings for item i when all users have rated i, $\sigma_q(i)$ the queue position of item i at node q, B average device contact capacity, λ average device contact rate, H_i set of devices with item i.

First, we present an equation to bound $P\{\Omega_i > g_i^+ + r_i^+\}$, the probability that the number of correct positive predictions for item i would increase if i is transferred, is given as follows:

$$Pr\{\Omega_i > r_i^+ + g_i^+\} \leq \min\left\{1, e^{\frac{r_i^+ E[\Omega_i]}{n - r_i^+}} \left(1 - \frac{r_i^+}{n}\right)^{r_i^+ + g_i^+}\right\} \tag{1}$$

For the ratings and items to be useful, the item should reach a user before some time deadline. Estimated probability of delivery an item i late after the time deadline t is:

$$Pr\{Y_i \geq t\} > 1 - \min\{1, \frac{|N_i|}{B\lambda t|H_i|} \sum_{v \in H_i} \sigma_{i,v}\} \tag{2}$$

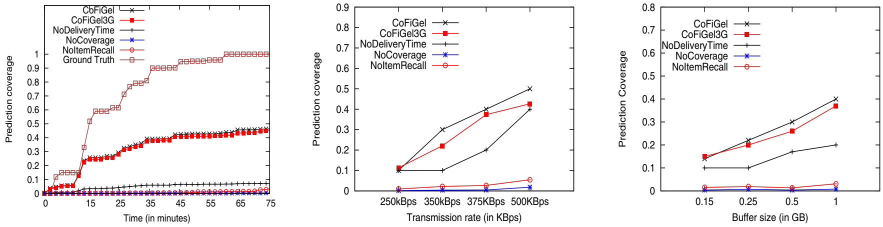
The proofs and discussions of Equations 1 and 2 are available in [1].

We now present the workings of CoFiGel based on Equation 1 and 2. At each device, CoFiGel decides which item to transmit by computing a utility U_i, which incorporates the number of positive ratings (rated or predicted) for i, the probability of gain in ratings, and the probability of delivery within the deadline: $U_i = (g_i^+ + r_i^+) \cdot G_i \cdot D_i$, where G_i is the right-hand-side of Equation 1, and D_i is the right-hand-side of Equation 2. The utility increases if either (i) the total number of correctly predicted positive ratings we get eventually $(g_i^+ + r_i^+)$, increases (ii) the likelihood of the number of correct predictions increases (G_i), or (iii) the likelihood of delivering an item within the deadline t increases. Note that since the bounds provided are very loose, we do not expect these computed utilities to reflect the true value of the rating gain. For scheduling, only relative ordering is important and we transfer items in decreasing order of utility.

Table 2. Simulation Parameters

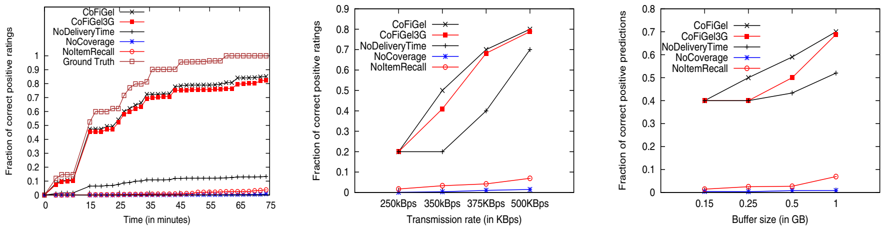
Parameter	RollerNet
Number of Publisher and Subscriber Nodes	10 and 30
(Item publisher rate)/publisher and item lifetime	40 items/Hr and 1 hour 15 min
Simulation duration, warmup and cool down time	Approx.3 Hrs, 1 Hr and 0.5 Hr
Item size and Buffer size	15MB and 1GB
Default contact bandwidth	3Mbps

5 Simulation Evaluation



(a) Prediction coverage over time (b) Prediction coverage vs. Transmission Rate (c) Prediction coverage vs. Buffer

Fig. 1. RollerNet trace (total ratings = 11536)



(a) Fraction of correct positive predictions over time (b) Fraction of correct positive predictions vs. Transmission Rate (c) Fraction of correct positive predictions vs. Buffer

Fig. 2. RollerNet trace (total positive ratings = 6400)

5.1 Simulation Setting

To evaluate *CoFiGel*, we use the MovieLens data set[4] as the underlying user ratings. The data set chosen has 100,000 ratings, 943 users and 1,682 items. We use the RollerNet trace ([6]) as the human mobility traces. It consists of about 60 Bluetooth devices carried

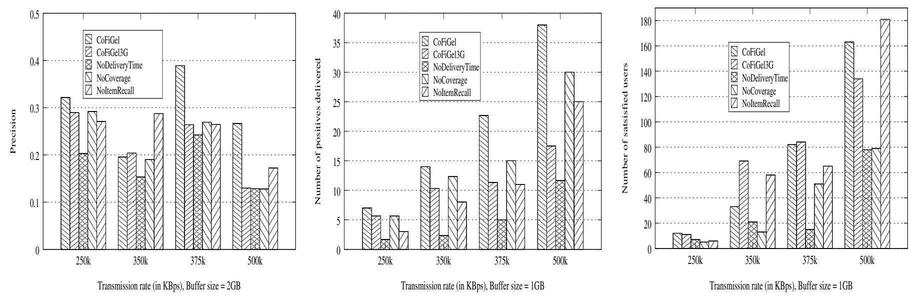
[4] `http://movielens.umn.edu`

by groups of roller bladers in a roller tour over three hours. The average contact duration is 22 seconds, with an average of 501 contacts made per node over 3 hours.

Video size of user generated content such as those found in popular sites like YouTube is 25MB or less (98% of videos are 25MB or less [4]). We choose data size of 15MB. The buffer size and item generation rate are similarly adjusted to ensure sufficient loading in the system. As some nodes in the trace have very limited contacts with the rest of the trace, we avoid selecting these nodes as the publisher or subscriber nodes (though they can still act as relay nodes). These nodes are identified as nodes which do not have sufficient number of node contacts and contact bandwidth to support meaningful data exchange. After removing these nodes, 10 publishers and 30 subscribers were chosen. In order to reduce simulation time, we reduce the MovieLens data set selected by randomly choosing 900 items (movies) and 500 users from the original data set. All user-item ratings associated with these chosen user-item pairs from the original dataset are also included. Finally, as the rating data set and the mobility trace are generated independently, we map the rating data to the mobility trace in the following way: Every item in the reduced data set is randomly assigned to a publisher node in the mobility trace. This node will act as the publisher for the item. Every user in the reduced data set is randomly mapped onto a mobile subscriber node. The actual user-item rating is known only when the item reaches the given mobile node where the user is located.

Table 2 are used as default unless otherwise specified. Each simulation point is run at least 3 times with different random seeds. The performance objectives used are prediction coverage, precision, recall and number of satisfied users and latency, as described in Section 3.

We compare the performance of *CoFiGel* with four other algorithms, namely: **(1)** A scheme that knows the ground-truth of data available. The ground-truth is available from the MovieLens data set. This scheme provides the actual rating coverage and gives an upper bound on the system performance. This scheme is used only in the coverage comparison since ground-truth is not applicable in the user satisfaction evaluation. **(2)** An epidemic-based algorithm that is similar to *CoFiGel* except that it does not take into account contact history and time constraints. We called this algorithm **NoDeliveryTime**. The performance difference between **NoDeliveryTime** and *CoFiGel* indicates the improvement provided by exploiting contact history. **(3)** An algorithm that uses only the rating information available. This is referred to as **NoCoverage**. The ratings of the items are predicted using the MCF, but the rating update and the potential coverage increase is not considered. By using only limited rating information, **NoCoverage** is expected to perform the worst. **(4)** An algorithm which tries to schedule an item so as to acquire prediction coverage of hitherto unrated users and to satisfy as many more users as possible. This is called the **NoItemRecall**. While this approach also uses contact history, it does not perform multi-round predictions as in the case of *CoFiGel*. It only acts using the current rating information. **(5) CoFiGel3G** is a modification of *CoFiGel* such that it uses the cellular network to upload/download ratings and a central server to run the MCF. However, the data are still sent over the DTN. By exploiting the cellular network as control channel, ratings information propagate quickly among the nodes and



(a) Transmission Rate Vs Precision (b) Transmission Rate Vs Delivered positive items (c) Transmission Rate Vs Users

Fig. 3. RollerNet trace

is always up-to-date. However, it is important to note that faster rating propagation does not always translate to higher rating coverage. This is because an actual rating can only be discovered after an user has access to the actual video and provides the rating.

5.2 Coverage

We now evaluate the performance of *CoFiGel* and the other algorithms in terms of prediction coverage, a commonly used metric for MCF. In addition, we also measured the fraction of correctly predicted positive (or FCPP) items, which measure the ratio of correctly predicted positive item to the total of number of positive ratings rather over all ratings. Given that we are simulating a DTN environment, FCPP provides a better gauge for what is achievable by good algorithms in more challenging environments.

Figures 1(a) and 2(a) show how positive ratings increase over time. The actual number of ratings for the items published so far (*Ground-Truth*) are shown to illustrate the best possible outcomes. In terms of overall ratings, CoFiGel discovers 45% of the ratings. In terms of FCPP, *CoFiGel* discovers 84% of the positive ratings. In fact, the performance of *CoFiGel* measured using FCPP closely matches the actual ratings in the first 15 minutes and the gap remains small throughout the simulation. The results shows that *CoFiGel* has the best performance, followed by *CoFiGel3G*.

This result can be somewhat surprising since *CoFiGel3G* uses the same algorithm as *CoFiGel* but uses the control (3G) channel for centralized rating computation and sharing. We explain the result as follows. Since *CoFiGel3G* performs centralized rating, the rating matrix gets updated much faster. This fast rating update has the (unintended) consequence that the variable G_i in utility function approaches the value of 1.0 much faster than the case for *CoFiGel*. As the value of G_i gets close to 1 and saturates around this value, this variable becomes useless in term of providing information for relative ranking to decide which video data item is more important. However, since propagation of video data item lags behind rating data, the loss of this rating information results in *CoFiGel3G* performing worse than *CoFiGel*.

The higher contact rate and capacity turn out to have adverse effect on **NoDeliveryTime**, **NoCoverage** and **NoItemRecall**, since each algorithm only looks at one aspect of the problem. In terms of FCPP, **NoDeliveryTime** discovers 13% of the positive ratings, while **NoCoverage** discovers 0.6% or less of the positive ratings and **NoItemRecall** discovers around 1%.

The coverage for the **NoCoverage** is very low, showing that it is important to take into account additional information beyond ratings. Figures 1(b) and 2(b) show how coverage varies with transmission rate. While increase in contact capacity results in increased coverage because more items get rated, *CoFiGel* is able to exploit the increase in transmission rate much better than **NoDeliveryTime**, **NoCoverage** and **NoItemRecall**. In the results shown, *CoFiGel* performs better than **NoDeliveryTime** by up to 105% and discovers at least 50 times more ratings than **NoCoverage** and **NoItemRecall** consistently. In general, more improvement comes from taking into account rating coverage gain (from **NoCoverage** to **NoDeliveryTime**) than taking into account contact history. The effort by **NoItemRecall** to increase the number of user ratings is also ineffective due to the absence of rating gain which is capitalized by *CoFiGel*. Nevertheless, substantial improvement is still observed between **NoDeliveryTime** and *CoFiGel*. The performance with respect to different buffer sizes is shown in Figures 1(c), 2(c). There are two observations. First, for very small buffer size of less than 150MB, very few items make it to the next hop and hence, the FCPP remains same for *CoFiGel* and **NoDeliveryTime**. For larger buffer sizes, FCPP of *CoFiGel* is higher than **NoDeliveryTime** by up to 36% and for **NoCoverage** by 50 to 60 times.

5.3 User Satisfaction

While coverage indicates the predictive power of the system, the actual user satisfaction has to be measured by looking at how many items reach users that like them. In order to ensure that the nodes have accumulated enough training data before making the measurement, we consider items generated after first 1.5 hours and before the last half hour. The first 1.5 hours serve as the training phase, while the last half hour is ignored to make sure that items generated later in the trace do not bias the measurement.

Figure 3(a) shows the results for precision of items reaching the users. It is clear that *CoFiGel* performs very well, except for one case (350K), it has the highest precision. In addition, note that even though **NoItemRecall** has a higher precision, from the results in the previous section, it has very low coverage. Due to the disconnected nature of DTN and the large number of data items and users available, it is also useful to look user utility in two other ways. First, we look at the average number of positively rated items that reach any user. The result is shown in figure 3(b). *CoFiGel* clearly outperforms the other two algorithms by a very large margin once the bandwidth exceeds some threshold required for data dissemination. At the highest transmission rate experimented, improvements are 117% compared to **NoDeliveryTime** and 225% more useful items than **NoCoverage**. Another way we measure recall is to look at the number of users who have received at least one useful item. The result is shown in figure 3(c). Again, *CoFiGel* performs well, in particular, at higher bandwidth. At 4Mbps, *CoFiGel* delivers twice as many useful items to users than **NoCoverage** and **NoDeliveryTime**.

6 Conclusion

We have presented *CoFiGel*, a novel approach that combines collaborative filtering and DTN routing in a distributed environment with intermittent connectivity. It is designed for sharing of locally stored contents that have *spatial and temporal relationships*. Results show *CoFiGel* ensures timely deliver of items with higher prediction coverage gain, discovers more ratings and delivers more items that are rated positively by users, than baseline strategies.

References

1. Seshadri, P.V., Chan, M.C., Ooi, W.T.: Mobile-to-Mobile Video Recommendation, (November 12, 2012), `http://arxiv.org/abs/1211.2063`
2. Huang, J., Xu, Q., Tiwana, B., Mao, Z.M., Zhang, M., Bahl, P.: Anatomizing Application Performance Differences on Smartphones. In: MobiSys, pp. 165–178 (2010)
3. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. In: Advances in Artificial Intelligence, vol. 2009. Hindawi Publishing Corp. (2009)
4. Xu, C., Cameron, D., Jiangchuan, L.: Statistics and Social Network of YouTube Videos. In: IWQoS, pp. 229–238 (2008)
5. Vidal, J.M.: A Protocol for a Distributed Recommender System. In: Falcone, R., Barber, S.K., Sabater-Mir, J., Singh, M.P. (eds.) Trusting Agents. LNCS (LNAI), vol. 3577, pp. 200–217. Springer, Heidelberg (2005)
6. Tournoux, P.U., Leguay, J., Benbadis, F., Conan, V., Dias de Amorim, M., Whitbeck, J.: The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing. In: INFOCOM, pp. 1116–1124 (2009)
7. Shlomo, B., Tsvi, K., Francesco, R.: Distributed collaborative filtering with domain specialization. In: RecSys, pp. 33–40 (2007)
8. Wang, J., Pouwelse, J., Lagendijk, R.L., Reinders, M.J.T.: Distributed collaborative filtering for peer-to-peer file sharing systems. In: SAC, pp. 1026–1030 (2006)
9. Balasubramanian, N., Balasubramanian, A., Venkataramani, A.: Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In: IMC, pp. 280–293 (2009)
10. Ioannidis, S., Chaintreau, A., Massoulie, L.: Optimal and Scalable Distribution of Content Updates over a Mobile Social Network. In: INFOCOM, pp. 1422–1430 (2009)
11. Lin, K.C.-J., Chen, C.-W., Chou, C.-F.: Preference-Aware Content Dissemination in Opportunistic Mobile Social Networks. In: INFOCOM, pp. 1960–1968 (2012)
12. Lo Giusto, G., Mashhadi, A.J., Capra, L.: Folksonomy-based reasoning for content dissemination in mobile settings. In: CHANTS, pp. 39–46 (2010)
13. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. In: Internet Computing, vol. 7, pp. 76–80. IEEE (2003)
14. Miller, B.N., Konstan, J.A., Riedl, J.: PocketLens: Toward a Personal Recommender System. In: Transactions on Information Systems, vol. 22, pp. 437–476. ACM (2004)
15. Lee, K., Yi, Y., Jeong, J., Won, H., Rhee, I., Chong, S.: Max-Contribution: On Optimal Resource Allocation in Delay Tolerant Networks. In: INFOCOM, pp. 1136–1144 (2010)